Exercises - IPO, Variables, and Constants

Once you are finished all of the exercises, upload all of your .py files. Be sure to include your header, use specific variable names and remember to comment each part of your code.

Key Concepts: (The problem set starts on page 2 but you should read the note below).

1. **Constants** are variables that have values that do not change. They are always entirely in uppercase characters and declared before variables at the beginning of your program.

   Example:

   HST= 0.13
   price = 34.56

2. You can convert a string into a float or an int.

Example:
price = float ("23.45") #variable price is of type float and has a numeric #value of 23.45

price = int ("23.45") # price now has the integer value of 23.

3. **round (123.4555, 2)** #returns 123.46. The first argument is the number you wish to round and the second input is the number of decimal places you wish to round to.

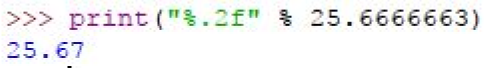
Another example:
round(12.201, 2) # this returns 12.2 (it cuts off the zero as it does not affect the overall value.

Exercises - IPO, Variables, and Constants

1. Write a program that asks the user for two prices of items and then a discount as a percentage. Make sure each variable has a specific name. Your program should calculate the *final price* for purchasing both items together after the discount and 13% HST has been applied. Apply the discount first and then the tax. Use a constant variable to store the HST value. Your final answer must be rounded to two decimal places. Remember that financial values are represented by floats rounded to two decimal places.

To format a print statement to show a float value rounded to two decimal places, you could use the line below. It is better to store the result in a variable, then you print using the variable name specifically.

If the cost works out to 34.50, the round(34.50, 2) will result in 34.5. You need to show the zero at the end so you will format it properly in your print statement as follows:



```
>>> print("%.2f" % 25.6666663)
25.67
```

Save your file as yourFirstName_IVC001.py

2. Write a program that asks the user for the radius and height of a cylinder. Your program should calculate the volume of the cylinder (rounded to the nearest whole number) and print it to screen (look up the formula on Google). Remember to use a constant (if there is one in the formula) and to declare it before regular variables. Constants are always fully capitalized (meaning that every letter is capitalized in the name) and declared before regular variables.
Save your program as yourFirstName_IVC002.py

3. Create a program the will prompt the user for an item description (string), the price of the item (float), and the quantity (int). The program will output the final cost including all taxes. Remember that you want the user to enter in their text on the next line (or new line) and not on the same line as the question. What do you have to add to your input line to accomplish this? You may need to review one of the PowerPoints if you do not know. Remember to include the lines with the ---- in your text and to

reduce the length of string literals. Do you really need to type out each dash in your print statement? What can you do instead?

Sample Output

```
Enter the description:
Volleyball
Enter the price:
39.99
Enter the quantity:
4
RECEIPT
------------------------------------------------
Description  QTY     price     Subtotal
Volleyball   4       $39.99    $159.96

HST:                          $20.80
------------------------------------------------
Total:                        $180.76
```

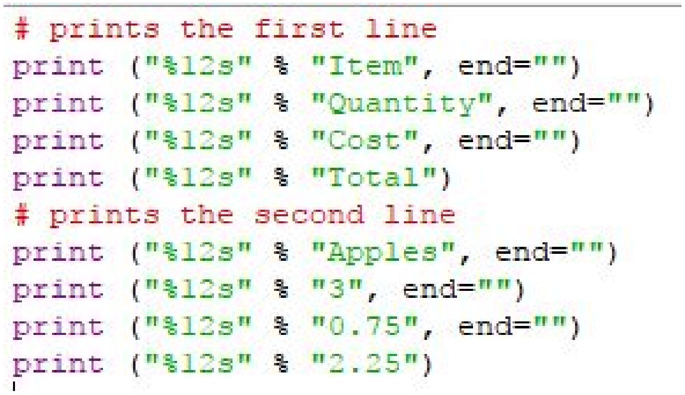
Save your program as yourFirstName_IVC003.py

4. Another way to format output is by using field widths.

NOTE:
A field width specification can be included to tell Python how much space you want something to use when printed on the screen. This is useful when you are printing columns of data or information, such as on a sales receipt. Type in the code below in script mode and notice how the output is formatted. You do not need to submit this part. This is for your own knowledge. The number 12 represents the width that is being allotted to displaying the result to the screen. The 's' represents that a string type is being formatted.

Example:

Exercises - IPO, Variables, and Constants

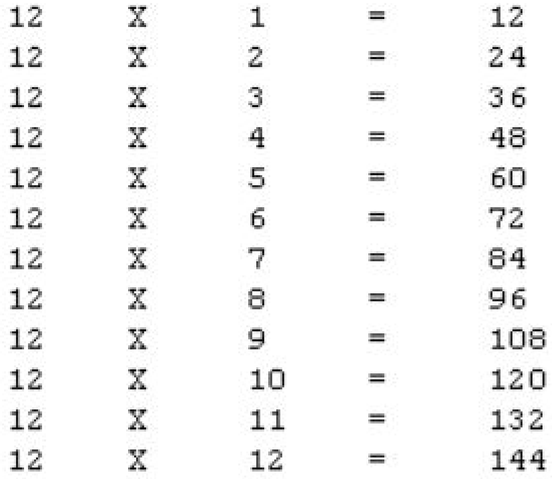


```
# prints the first line
print ("%12s" % "Item", end="")
print ("%12s" % "Quantity", end="")
print ("%12s" % "Cost", end="")
print ("%12s" % "Total")
# prints the second line
print ("%12s" % "Apples", end="")
print ("%12s" % "3", end="")
print ("%12s" % "0.75", end="")
print ("%12s" % "2.25")
```

Note: In **Python version 3.x**, you can use the `end` argument to the `print()` function to prevent a newline character from being printed. This means that print statements on separate lines can be printed on the same line in the output.

Here is the part you need to program:
Create a neatly-spaced "times table" from 12 x 1 to 1 x 12 that uses field widths. Use formulas to calculate the products.

Example Output:



```
12      X       1       =       12
12      X       2       =       24
12      X       3       =       36
12      X       4       =       48
12      X       5       =       60
12      X       6       =       72
12      X       7       =       84
12      X       8       =       96
12      X       9       =       108
12      X       10      =       120
12      X       11      =       132
12      X       12      =       144
```

Save your program as **yourFirstName_IVC004.py**